瞭解Nexus交換機上的循環冗餘檢查錯誤

目錄

[簡介](#)
[必要條件](#)
[需求](#)
[採用元件](#)
[背景資訊](#)
[適用硬體](#)
[CRC定義](#)
[CRC錯誤定義](#)
[CRC錯誤的常見症狀](#)
[在Windows主機上接收的錯誤](#)
[Linux主機上的RX錯誤](#)
[網路裝置上的CRC錯誤](#)
[儲存轉發網路裝置上的輸入錯誤](#)
[直通網路裝置上的輸入和輸出錯誤](#)
[跟蹤和隔離CRC錯誤](#)
[CRC錯誤的根本原因](#)
[解決CRC錯誤](#)
[相關資訊](#)

簡介

本檔案將詳細介紹有關在介面計數器上觀察到的循環冗餘檢查(CRC)錯誤以及Cisco Nexus交換器的統計資訊。

必要條件

需求

思科建議您瞭解乙太網交換和Cisco NX-OS命令列介面(CLI)的基本知識。 如需詳細資訊，請參閱以下適用檔案之一：

- Cisco Nexus 9000 NX-OS基礎配置指南10.2(x)版
- Cisco Nexus 9000系列NX-OS基礎配置指南9.3(x)版
- Cisco Nexus 9000系列NX-OS基礎配置指南9.2(x)版
- Cisco Nexus 9000系列NX-OS基礎配置指南7.x版

- 乙太網故障排除

採用元件

本文中的資訊係根據以下軟體和硬體版本：

- 從NX-OS軟體版本9.3(8)開始的Nexus 9000系列交換機
- 從NX-OS軟體版本9.3(8)開始的Nexus 3000系列交換機

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

背景資訊

本檔案將詳細介紹在Cisco Nexus系列交換器上的介面計數器上觀察到的循環冗餘檢查(CRC)錯誤。本檔案將說明CRC是什麼、如何在乙太網幀的幀校驗序列(FCS)欄位中使用它、CRC錯誤如何在Nexus交換機上顯示、CRC錯誤如何在儲存轉發交換和直通交換場景中互動、CRC錯誤最可能的根本原因，以及如何排除和解決CRC錯誤。

適用硬體

本檔案中的資訊適用於所有Cisco Nexus系列交換器。本檔案中的某些資訊也可適用於其他Cisco路由和交換平台，例如Cisco Catalyst路由器和交換器。

CRC定義

CRC是一種常見的錯誤檢測機制，用於識別在傳輸過程中更改或損壞的資料。當連線到網路的裝置需要傳輸資料時，該裝置會針對導致固定長度數字的資料運行基於循環代碼的計算演算法。此固定長度數字稱為CRC值，但在口語中，它通常簡稱為CRC。此CRC值附加在資料上，並通過網路傳輸到另一裝置。該遠端裝置對資料運行相同的循環碼演算法，並將結果值與附加在資料上的CRC進行比較。如果兩個值都匹配，則遠端裝置假定資料通過網路傳輸，而不會損壞。如果值不匹配，則遠端裝置會假設資料在網路傳輸期間已損壞。無法信任此損壞的資料，因此將其丟棄。

CRC用於在多種電腦聯網技術中檢測錯誤，例如乙太網（有線和無線兩種變體）、令牌環、非同步傳輸模式(ATM)和幀中繼。乙太網幀在插入32位CRC值的幀末尾具有32位幀校驗序列(FCS)欄位。

例如，假設兩個名為Host-A和Host-B的主機通過其網路介面卡(NIC)直接連線。主機A需要通過網路向主機B傳送句子「這是一個示例」。主機A製作一個乙太網幀，該幀的目的地址是主機B，其負載為「這是一個示例」，並且計算出幀的CRC值是十六進位制值0xABCD。Host-A將CRC值0xABCD插入乙太網幀的FCS欄位，然後將乙太網幀從主機A的NIC傳輸到主機B。

主機B收到此訊框後，會使用與主機A完全相同的演演算法來計算訊框的CRC值。Host-B計算幀的CRC值為0xABCD的十六進位制值，它向Host-B指示在將幀傳輸到主機B時，乙太網幀未損壞。

CRC錯誤定義

當裝置（網路裝置或連線到網路的主機）接收到乙太網幀時，該幀的FCS欄位中的CRC值與裝置為幀計算得出的CRC值不匹配，就會發生CRC錯誤。

這個概念最好通過示例來說明。考慮以下情況：名為Host-A和Host-B的兩台主機通過其網路介面卡(NIC)直接相連。 主機A需要透過網路將句子「This is an example」傳送給主機B。主機A製作一個乙太網幀，該幀的目的地址是主機B，其負載為「這是一個示例」，並且計算出該幀的CRC值是十

六進位制值0xABCD。Host-A將CRC值0xABCD插入乙太網幀的FCS欄位，然後將乙太網幀從主機A的NIC傳輸到主機B。

但是，將主機A連線到主機B的物理介質上的損壞會損壞幀的內容，使得幀內的句子變為「This was an example」，而不是期望的有效負載「This is an example」。

主機B收到此幀時，將計算包含損壞的負載的幀的CRC值。主機B計算幀的CRC值為0xDEAD的十六進位制值，這與乙太網幀的FCS欄位中的0xZHI CRC值不同。CRC值的這種差異告知主機B在將乙太網幀傳輸到主機B時，該乙太網幀已損壞。因此，主機B無法信任該乙太網幀的內容，因此會丟棄該幀。主機B通常也會在其網路介面卡(NIC)上增加某種錯誤計數器，例如「input errors」、「CRC errors」或「RX errors」計數器。

CRC錯誤的常見症狀

CRC錯誤通常通過以下兩種方式之一表現出來：

1. 網路連線裝置介面上的遞增或非零錯誤計數器。
2. 由於網路連線裝置丟棄損壞的幀，導致通過網路的流量丟包/幀丟失。

這些錯誤表現方式略有不同，具體取決於您使用的裝置。以下各小節針對每種裝置進行了詳細介紹。

在Windows主機上接收的錯誤

Windows主機上的CRC錯誤通常顯示為**netstat -e**命令輸出中顯示的非零**已接收錯誤**計數器。Windows主機命令提示符中的非零已接收錯誤計數器的示例如下：

```
>netstat -e
Interface Statistics

                           Received            Sent
Bytes                    1116139893      3374201234
Unicast packets           101276400        49751195
Non-unicast packets               0               0
Discards                          0               0
Errors                        47294               0
Unknown protocols                 0
```

NIC及其各自的驅動程式必須支援對NIC接收到的CRC錯誤進行記帳，以便使**netstat -e**命令報告的已接收錯誤數準確。大多數現代NIC及其各自的驅動程式都支援對NIC接收的CRC錯誤進行準確的記帳。

Linux主機上的RX錯誤

Linux主機上的CRC錯誤通常顯示為**ifconfig**命令輸出中顯示的非零「RX錯誤」計數器。來自Linux主機的非零RX錯誤計數器的示例如下：

```
$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.0.2.10  netmask 255.255.255.128  broadcast 192.0.2.255
       inet6 fe80::10  prefixlen 64  scopeid 0x20<link>
       ether 08:62:66:be:48:9b  txqueuelen 1000  (Ethernet)
       RX packets 591511682  bytes 214790684016 (200.0 GiB)
```

```
        RX errors 478920  dropped 0  overruns 0  frame 0
       TX packets 85495109  bytes 288004112030 (268.2 GiB)
       TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Linux主機上的CRC錯誤也可能顯示為**ip -s link show**命令輸出中顯示的非零「RX錯誤」計數。來自Linux主機的非零RX錯誤計數器的示例如下：

```
$ ip -s link show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group
default qlen 1000
   link/ether 08:62:66:84:8f:6d brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped overrun mcast
   32246366102 444908978 478920       647     0       419445867
   TX: bytes  packets  errors  dropped carrier collsns
   3352693923 30185715 0       0       0       0
   altname enp11s0
```

NIC及其各自的驅動程式必須支援對NIC接收的CRC錯誤進行記帳，以便由**ifconfig**或**ip -s link show**命令報告的RX錯誤數量準確。大多數現代NIC及其各自的驅動程式都支援對NIC接收的CRC錯誤進行準確的記帳。

網路裝置上的CRC錯誤

網路裝置運行在兩種轉發模式之一 — 儲存轉發模式和直通轉發模式。網路裝置處理接收的CRC錯誤的方式因轉發模式而異。此處的子部分將描述每種轉發模式的具體行為。

儲存轉發網路裝置上的輸入錯誤

當在儲存轉發模式下運行的網路裝置收到幀時，網路裝置將在您驗證幀的CRC值之前緩衝整個幀（「儲存」），對幀做出轉發決策，並將幀從介面傳送出去（「轉發」）。 因此，當在儲存轉發模式下運行的網路裝置收到特定介面上包含錯誤CRC值的損壞幀時，它會丟棄該幀，並增加介面上的「輸入錯誤」計數器。

換句話說，損壞的乙太網幀不會由運行在儲存轉發模式下的網路裝置轉發；入口上會捨棄。

Cisco Nexus 7000和7700系列交換機在儲存轉發模式下運行。以下是Nexus 7000或7700系列交換機的非零輸入錯誤計數器和非零CRC/FCS計數器的示例：

```
switch# show interface
<snip>
Ethernet1/1 is up
 RX
   241052345 unicast packets  5236252 multicast packets  5 broadcast packets
   245794858 input packets  17901276787 bytes
   0 jumbo packets  0 storm suppression packets
   0 runts  0 giants  579204 CRC/FCS  0 no buffer
   579204 input error  0 short frame  0 overrun   0 underrun  0 ignored
   0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
   0 input with dribble  0 input discard
   0 Rx pause
```

CRC錯誤也可能在**show interface counters errors**的輸出中顯示為非零的「FCS-Err」計數器。此命令輸出中的「Rcv-Err」計數器也將具有非零值，該值是介面接收的所有輸入錯誤（CRC或其他）的總和。示例如下：

```
switch# show interface counters errors
```

```
<snip>
--------------------------------------------------------------------------------
Port          Align-Err    FCS-Err   Xmit-Err    Rcv-Err  UnderSize OutDiscards
--------------------------------------------------------------------------------
Eth1/1                0     579204          0     579204          0           0
```

直通網路裝置上的輸入和輸出錯誤

當以直通轉發模式運行的網路裝置開始接收幀時，網路裝置將對幀的報頭作出轉發決定，並在收到足夠的資料幀後立即開始從介面傳送幀，以便做出有效的轉發決定。由於幀和資料包報頭位於幀的開頭，因此通常會在收到幀的負載之前做出轉發決策。

乙太網幀的FCS欄位位於幀的末尾，緊跟幀的負載之後。因此，在直通轉發模式下運行的網路裝置在能夠計算幀的CRC時已經開始從另一個介面傳送幀。如果網路裝置為幀計算出的CRC與FCS欄位中的CRC值不匹配，則意味著網路裝置將損壞的幀轉發到網路中。發生這種情況時，網路裝置將增加兩個計數器：

1. 最初收到損壞幀的介面上的「輸入錯誤」計數器。
2. 傳輸損壞幀的所有介面上的「輸出錯誤」計數器。對於單播流量，這通常是單個介面 — 但是，對於廣播、組播或未知單播流量，這可以是一個或多個介面。

這裡顯示了一個範例，其中**show interface**命令的輸出表示網路裝置的Ethernet1/1上收到多個損毀的訊框，且由於網路裝置的直通轉送模式而從Ethernet1/2轉發出去：

```
switch# show interface
<snip>
Ethernet1/1 is up
 RX
   46739903 unicast packets  29596632 multicast packets  0 broadcast packets
   76336535 input packets  6743810714 bytes
   15 jumbo packets  0 storm suppression bytes
   0 runts  0 giants  47294 CRC  0 no buffer
   47294 input error  0 short frame  0 overrun   0 underrun  0 ignored
   0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
   0 input with dribble  0 input discard
   0 Rx pause

Ethernet1/2 is up
 TX
   46091721 unicast packets  2852390 multicast packets  102619 broadcast packets
   49046730 output packets  3859955290 bytes
   50230 jumbo packets
   47294 output error  0 collision  0 deferred  0 late collision
   0 lost carrier  0 no carrier  0 babble  0 output discard
   0 Tx pause
```

在**show interface counters** errors的輸出中，CRC錯誤還會在輸入介面上表現為非零「FCS-Err」計數器，而在輸出介面上表現為非零「Xmit-Err」計數器。此命令輸出中，輸入介面上的「Rcv-Err」計數器也將具有非零值，該值是該介面接收的所有輸入錯誤（CRC或其他）的總和。示例如下：

```
switch# show interface counters errors
<snip>
--------------------------------------------------------------------------------
Port          Align-Err    FCS-Err   Xmit-Err    Rcv-Err  UnderSize OutDiscards
--------------------------------------------------------------------------------
Eth1/1                0      47294          0      47294          0           0
Eth1/2                0          0      47294          0          0           0
```

網路裝置還將以特定的方式修改幀的FCS欄位中的CRC值，這表示上游網路裝置此幀已損壞。這種

行為稱為「堆積」CRC。修改CRC的準確方式因平台而異，但一般情況下，它涉及反轉幀的FCS欄位中存在的當前CRC值。以下是範例：

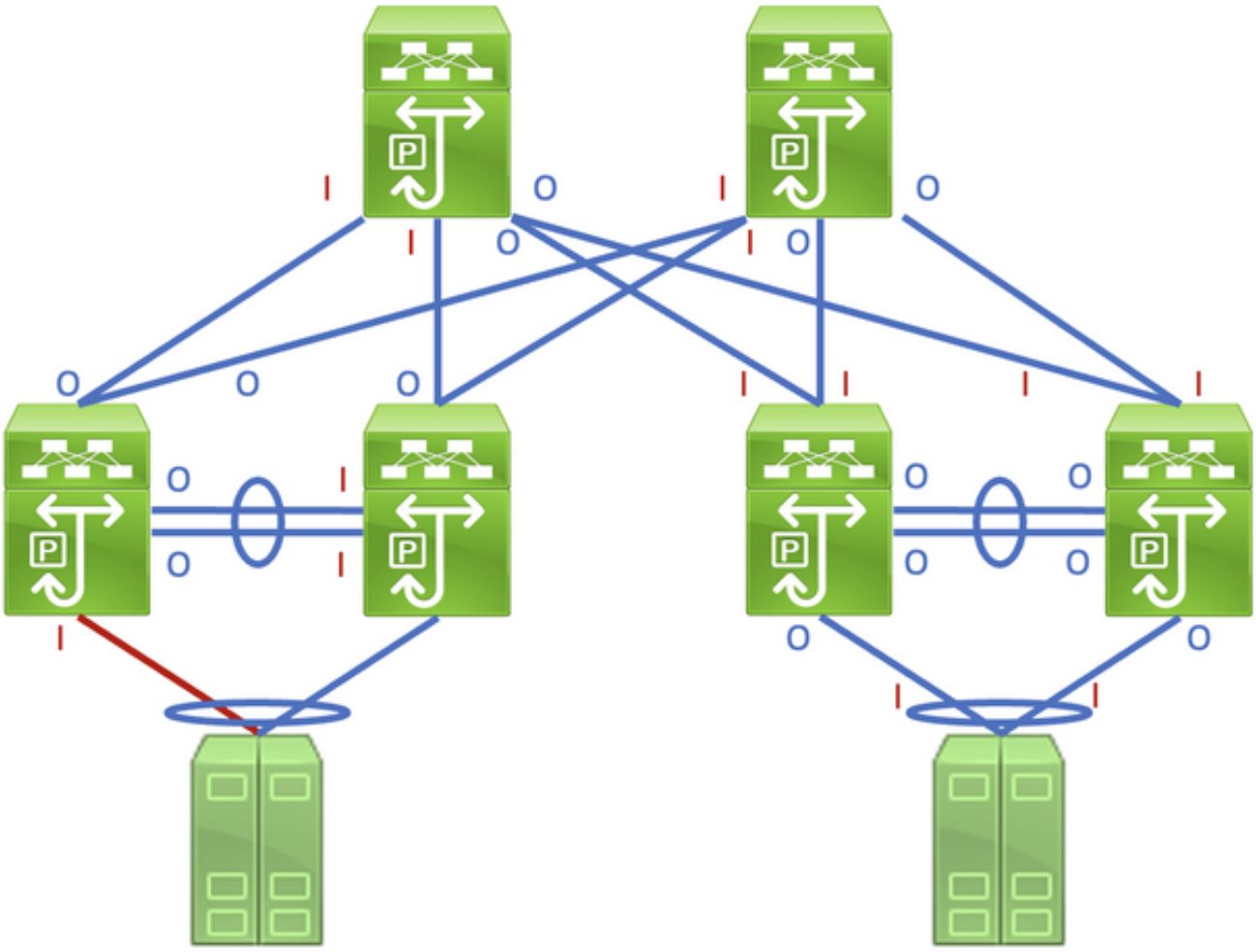
```
Original CRC: 0xABCD (1010101111001101)
Stomped CRC:  0x5432 (0101010000110010)
```

因此，在直通轉發模式下運行的網路裝置可以在整個網路中傳播損壞的幀。如果網路由在直通轉發模式下運行的多個網路裝置組成，則單個損壞的幀會導致網路中多個網路裝置上的輸入錯誤和輸出錯誤計數器增加。

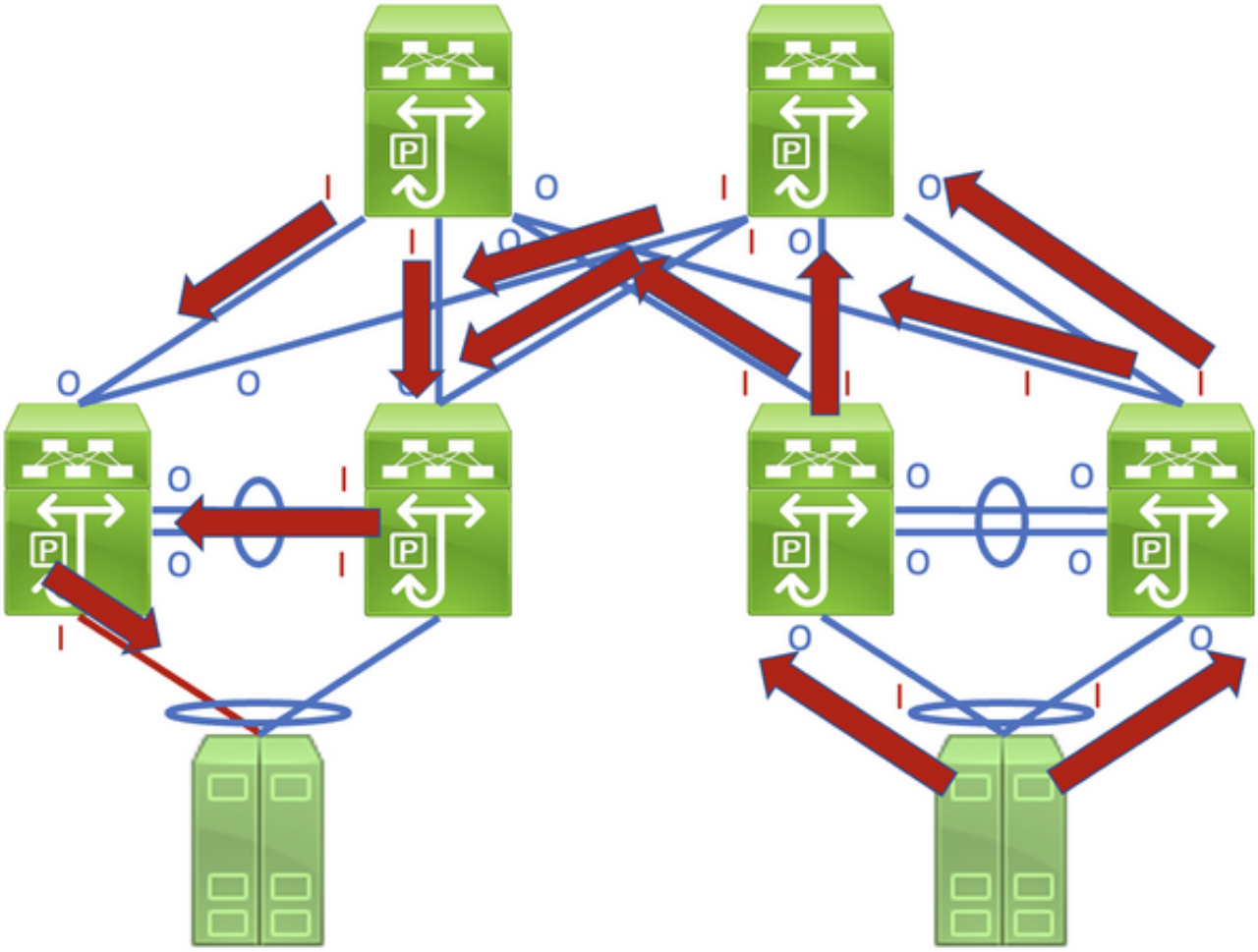
跟蹤和隔離CRC錯誤

要確定並解決CRC錯誤的根本原因，第一步是將CRC錯誤的來源隔離到網路中兩台裝置之間的特定鏈路。連線到此連結的一個裝置的介面輸出錯誤計數器值為零或不遞增，而連線到此連結的另一裝置的介面輸入錯誤計數器值為非零或遞增。這表示從一裝置完整介面流出的流量在向遠端裝置傳輸時已損壞，鏈路上另一裝置的輸入介面將其視為輸入錯誤。

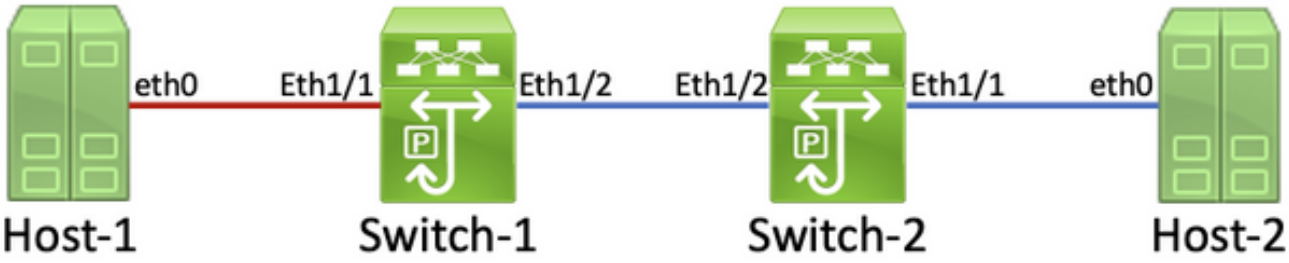
在由以儲存轉發模式運行的網路裝置組成的網路中識別此鏈路是一項簡單的任務。但是，在由以直通轉發模式運行的網路裝置組成的網路中標識此鏈路比較困難，因為許多網路裝置將具有非零輸入和輸出錯誤計數器。此處的拓撲中可看到此現象的示例，其中以紅色突出顯示的鏈路已損壞，因此經過該鏈路的流量已損壞。標有紅色「I」的介面表示可能有非零輸入錯誤的介面，而標有藍色「O」的介面表示可能有非零輸出錯誤的介面。



識別有故障的鏈路要求您通過非零輸入和輸出錯誤計數器遞迴地跟蹤網路中受損幀後面的「路徑」，非零輸入錯誤指向網路中受損鏈路的上游。這一點在此處的圖表中進行了演示。



通過示例最好地演示了用於跟蹤和識別損壞鏈路的詳細過程。考量以下拓撲：



在此拓撲中，名為Switch-1的Nexus交換機的介面Ethernet1/1通過Host-1的網路介面卡(NIC)eth0連線到名為Host-1的主機。Switch-1的介面Ethernet1/2通過Switch-2的介面Ethernet1/2連線到名為Switch-2的第二台Nexus交換機。Switch-2的介面Ethernet1/1通過Host-2的NIC eth0連線到名為Host-2的主機。

通過Switch-1的Ethernet1/1介面的Host-1和Switch-1之間的鏈路損壞，導致通過該鏈路的流量間歇性損壞。但是，我們還不知道此連結已損壞。我們必須通過非零或遞增輸入和輸出錯誤計數器跟蹤損壞幀在網路中離開的路徑，以定位此網路中損壞的鏈路。

在本例中，主機2的NIC報告其接收CRC錯誤。

```
Host-2$ ip -s link show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group
default qlen 1000
   link/ether 00:50:56:84:8f:6d brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped overrun mcast
   32246366102 444908978 478920       647     0       419445867
   TX: bytes  packets  errors  dropped carrier collsns
   3352693923 30185715 0       0       0       0
   altname enp11s0
```

您知道Host-2的NIC通過介面Ethernet1/1連線到Switch-2。您可以使用**show interface**命令確認介面Ethernet1/1具有非零輸出錯誤計數器。

```
Switch-2# show interface
<snip>
Ethernet1/1 is up
admin state is up, Dedicated Interface
    RX
      30184570 unicast packets  872 multicast packets  273 broadcast packets
      30185715 input packets  3352693923 bytes
      0 jumbo packets  0 storm suppression bytes
      0 runts  0 giants 0 CRC  0 no buffer
      0 input error  0 short frame  0 overrun   0 underrun  0 ignored
      0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
      0 input with dribble  0 input discard
      0 Rx pause
    TX
      444907944 unicast packets  932 multicast packets  102 broadcast packets
      444908978 output packets  32246366102 bytes
      0 jumbo packets
      478920 output error  0 collision  0 deferred  0 late collision
      0 lost carrier  0 no carrier  0 babble  0 output discard
      0 Tx pause
```

由於介面Ethernet1/1的輸出錯誤計數器不是零，因此交換器2的另一個介面很可能有非零輸入錯誤計數器。您可以使用**show interface counters errors non-zero**命令來識別Switch-2的任何介面是否有非零輸入錯誤計數器。

```
Switch-2# show interface counters errors non-zero
<snip>
--------------------------------------------------------------------------------
Port          Align-Err    FCS-Err   Xmit-Err    Rcv-Err  UnderSize OutDiscards
--------------------------------------------------------------------------------
Eth1/1                0          0     478920          0          0           0
Eth1/2                0     478920          0     478920          0           0

--------------------------------------------------------------------------------
Port         Single-Col  Multi-Col   Late-Col  Exces-Col  Carri-Sen       Runts
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
Port          Giants SQETest-Err Deferred-Tx IntMacTx-Er IntMacRx-Er Symbol-Err
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
Port         InDiscards
--------------------------------------------------------------------------------
```

您可以看到Switch-2的Ethernet1/2有非零輸入錯誤計數器。這表示Switch-2在此介面上收到損毀的流量。您可以通過Cisco Discovery Protocol(CDP)或Link Local Discovery Protocol(LLDP)功能確認

哪個裝置連線到Switch-2的Ethernet1/2。這裡使用**show cdp neighbors**命令顯示一個範例。

```
Switch-2# show cdp neighbors
<snip>
    Capability Codes: R - Router, T - Trans-Bridge, B - Source-Route-Bridge
    S - Switch, H - Host, I - IGMP, r - Repeater,
    V - VoIP-Phone, D - Remotely-Managed-Device,
    s - Supports-STP-Dispute

Device-ID          Local Intrfce  Hldtme Capability  Platform      Port ID
Switch-1(FDO12345678)
                    Eth1/2         125    R S I s   N9K-C93180YC- Eth1/2
```

現在您知道，Switch-2在其Ethernet1/2介面上收到來自Switch-1的Ethernet1/2介面的損壞流量，但您仍不知道Switch-1的Ethernet1/2與Switch-2的Ethernet1/2之間的鏈路是否損壞並導致損壞，或者交換機–1是轉發所收到損壞流量的直通交換機。您必須登入到Switch-1進行驗證。

您可以使用**show interfaces**指令確認Switch-1的Ethernet1/2介面具有非零輸出錯誤計數器。

```
Switch-1# show interface
<snip>
Ethernet1/2 is up
admin state is up, Dedicated Interface
    RX
      30581666 unicast packets  178 multicast packets  931 broadcast packets
      30582775 input packets  3352693923 bytes
      0 jumbo packets  0 storm suppression bytes
      0 runts  0 giants 0 CRC  0 no buffer
      0 input error  0 short frame  0 overrun   0 underrun  0 ignored
      0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
      0 input with dribble  0 input discard
      0 Rx pause
    TX
      454301132 unicast packets  734 multicast packets  72 broadcast packets
      454301938 output packets  32246366102 bytes
      0 jumbo packets
      478920 output error  0 collision  0 deferred  0 late collision
      0 lost carrier  0 no carrier  0 babble  0 output discard
      0 Tx pause
```

您可以看到Switch-1的Ethernet1/2有一個非零輸出錯誤計數器。這表示交換器1的Ethernet1/2和Switch-2的Ethernet1/2之間的連結沒有損毀，相反，Switch-1是中繼交換器，會轉送它在某些其他介面上接收的損毀流量。正如先前在Switch-2中所演示的那樣，您可以使用**show interface counters errors non-zero**命令來識別Switch-1的任何介面是否具有非零輸入錯誤計數器。

```
Switch-1# show interface counters errors non-zero
<snip>
--------------------------------------------------------------------------------
Port          Align-Err    FCS-Err   Xmit-Err    Rcv-Err  UnderSize OutDiscards
--------------------------------------------------------------------------------
Eth1/1                0     478920          0     478920          0           0
Eth1/2                0          0     478920          0          0           0

--------------------------------------------------------------------------------
Port         Single-Col  Multi-Col   Late-Col  Exces-Col  Carri-Sen       Runts
```

```
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
Port          Giants SQETest-Err Deferred-Tx IntMacTx-Er IntMacRx-Er Symbol-Err
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
Port         InDiscards
--------------------------------------------------------------------------------
```

您可以看到Switch-1的Ethernet1/1有非零輸入錯誤計數器。這表示Switch-1正在此介面上接收損毀的流量。我們知道該介面連線到Host-1的eth0 NIC。我們可以檢視Host-1的eth0 NIC介面統計資訊，以確認Host-1是否將損壞的幀從該介面傳送出去。

```
Host-1$ ip -s link show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group
default qlen 1000
   link/ether 00:50:56:84:8f:6d brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors dropped overrun mcast
   73146816142 423112898 0       0     0       437368817
   TX: bytes  packets  errors  dropped carrier collsns
   3312398924 37942624 0       0       0       0
   altname enp11s0
```

Host-1的eth0 NIC統計資訊表明主機沒有傳輸損壞的流量。這表示主機1的eth0和交換機1的Ethernet1/1之間的鏈路已損壞，是此流量損壞的來源。需要對此連結執行進一步的故障排除，找出導致此損壞的故障元件並更換它。

CRC錯誤的根本原因

CRC錯誤最常見的根本原因是兩台裝置之間的物理鏈路損壞或故障。示例包括：

- 物理介質（銅纜或光纖）或直接連線電纜(DAC)發生故障或損壞。
- 收發器/光纖發生故障或損壞。
- 配線面板埠故障或損壞。
- 網路裝置硬體故障（包括特定埠、線卡專用積體電路[ASIC]、媒體訪問控制[MAC]、交換矩陣模組等）、
- 主機中插入的網路介面卡出現故障。

一個或多個配置錯誤的裝置也可能在網路中無意中導致CRC錯誤。其中一個範例是網路中兩個或多個裝置之間的最大傳輸單元(MTU)組態不相符，導致大型封包被錯誤截斷。識別和解決此配置問題也可以糾正網路中的CRC錯誤。

解決CRC錯誤

您可以通過消除過程識別特定故障元件：

1. 使用同類已知良好的物理介質更換物理介質（銅介質或光纖）或DAC。
2. 將插入一個裝置介面的收發器更換為相同型號的已知良好收發器。如果這不能解決CRC錯誤，請使用同一型號的已知良好的收發器替換插入另一裝置介面的收發器。
3. 如果任何配線面板都用作損壞鏈路的一部分，請將鏈路移至配線面板上已知良好的埠。或者，如果可能，無需使用配線面板，即可通過連線鏈路來消除配線面板作為潛在的根本原因。

4. 將損壞的鏈路移至每台裝置上另一個已知良好的埠。您需要測試多個不同的埠，以隔離MAC、ASIC或線卡故障。
5. 如果損壞的鏈路涉及主機，請將該鏈路移至主機上的其他網絡卡。或者，將損壞的鏈路連線到已知良好的主機，以隔離主機網絡卡故障。

如果故障元件是思科產品（如思科網路裝置或收發器），並且受有效支援合約的保護，則您可以通過Cisco TAC開啟支援案例，詳細說明您的故障排除，以便通過退貨授權(RMA)更換故障元件。

相關資訊

- Nexus 9000雲端規模ASIC CRC識別與追蹤程式
- 技術支援與文件 - Cisco Systems